

Please type a plus sign (+) inside this box → ☐

PTO/SB/05 (4/98)
Approved for use through 09/30/2000. OMB 0651-0032
Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))

Attorney Docket No. 2207/8613

First Inventor or Application Identifier Jonathan P. DOUGLAS

Title CLOCK THROTTLE FOR INSTRUCTION PIPE

Express Mail Label No.

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. ☒ * Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)
2. ☒ Specification [Total Pages 15]
(preferred arrangement set forth below)
 - Descriptive title of the Invention
 - Cross References to Related Applications
 - Statement Regarding Fed sponsored R & D
 - Reference to Microfiche Appendix
 - Background of the Invention
 - Brief Summary of the Invention
 - Brief Description of the Drawings (if filed)
 - Detailed Description
 - Claim(s)
 - Abstract of the Disclosure
3. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 3]
4. Oath or Declaration [Total Pages 2]
 - a. ☒ Newly executed (original or copy)
 - b. ☐ Copy from a prior application (37 C.F.R. § 1.63(d))
(for continuation/divisional with Box 16 completed)
 - i. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting
inventor(s) named in the prior application,
see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).

* NOTE FOR ITEMS 1 & 13 IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

5. ☐ Microfiche Computer Program (Appendix)
6. Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)
 - a. ☐ Computer Readable Copy
 - b. ☐ Paper Copy (identical to computer copy)
 - c. ☐ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

7. ☒ Assignment Papers (cover sheet & document(s))
8. ☐ 37 C.F.R. § 3.73(b) Statement of Power of Attorney
(when there is an assignee)
9. ☐ English Translation Document (if applicable)
10. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
11. ☐ Preliminary Amendment
12. ☒ Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)
13. ☐ * Small Entity Statement(s) ☐ Statement filed in prior application
(PTO/SB/09-12) Status still proper and desired
14. ☐ Certified Copy of Priority Document(s)
(if foreign priority is claimed)
15. ☐ Other: _____

16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP)

of prior application No. _____ / _____

Prior application information: Examiner _____

Group / Art Unit: _____

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. CORRESPONDENCE ADDRESS

☐ Customer Number or Bar Code Label

(Insert Customer Number or Bar code label here)

or ☒ Correspondence address below

Name	23838				
Address	PATENT TRADEMARK OFFICE				
City	State	Zip Code			
Country	Telephone	Fax			

Name (Print/Type)	Robert L. Hails, Jr.	Registration No. (Attorney/Agent)	39,702
Signature		Date	6/21/2000

Burden Hour Statement This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

2207/8613
Patent

United States Patent Application
for

CLOCK THROTTLE FOR INSTRUCTION PIPE

Inventor:

Jonathan P. Douglas

PREPARED BY:

KENYON & KENYON
1500 K ST., N.W.
WASHINGTON, D.C. 20005

(202) 220-4200

CLOCK THROTTLE FOR INSTRUCTION PIPE

BACKGROUND

The present invention relates to a control mechanism for an instruction pipe in a processor that maintains timing synchronism between the instruction pipe and other elements outside the instruction pipe.

Execution logic in modern processors have begun to incorporate multiple instruction pipes. Each instruction pipe may include sufficient circuitry to execute most program instructions independently of the other pipes. Thus, a processor having multiple instruction pipes may perform nearly perfect parallel execution of program instructions.

It may not be desirable for instruction pipes to operate with complete independence from each other. For certain operations, greater efficiencies may be achieved by having the instruction pipes share access to other logic circuits. By way of example, it may be preferable for multiple instruction pipes to share a single Return Stack Buffer ("RSB"). As is known, an RSB is a buffer that stores forward and return pointers associated with call and return instructions. When a processor executes a call, it pushes an address associated with the call instruction to the RSB, typically the address of an instruction immediately following the call instruction, and begins execution at another program instruction at an address specified in the body of the call instruction. When a processor executes a return instruction, it retrieves an address from the top of the RSB and commences program execution at the retrieved address. Even in a processor having multiple instruction pipes, it may be more efficient to provide a single RSB for all instruction pipes rather than to provide a separate RSB for each of the instruction pipes. Because RSBs typically are not used every clock cycle, sharing the RSB improves utilization and reduces cost over a double-RSB design, for example.

In one implementation, an RSB may be provided within a first instruction pipe. Other instruction pipes in the processor may communicate with the RSB to store addresses therein for call instructions or to retrieve addresses therefrom for return instructions. However, this implementation raises a variety of timing problems.

A first timing problem arises because one RSB must be shared among a variety of instruction pipes. For N instruction pipes in a processor, each instruction pipe may enjoy utilization of the RSB reduced on a pro rata basis ($1/N^{\text{th}}$ of the RSB's total capacity). If an

instruction pipe issues read or write requests to the RSB in excess of its pro rata share, the requests may be dropped. This would result in processor failure.

A second timing problem may arise due to round-trip communication latencies between an instruction pipe and the RSB. Requests must propagate from an instruction pipe to an RSB, be acted upon by the RSB and results therefrom must return to the instruction pipe. An instruction pipe that does not account for this round-trip latency during operation may act upon invalid data. Again, this would result in processor failure.

Accordingly, there is a need in the art for a timing control mechanism for use in instruction pipes to conform operation of the instruction pipe to timing limitations that may arise when interfacing the instruction pipe with external elements.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of clock throttling logic according to an embodiment of the present invention.

FIG. 2 is a state diagram illustrating operation of a state machine according to an embodiment of the present invention.

FIG. 3 is a block diagram illustrating a dual-pipe architecture for a processor.

DETAILED DESCRIPTION

Embodiments of the present invention provide a clock throttling mechanism for a dual instruction pipe processor. In such an embodiment, an external element such as an RSB may be shared among a plurality of instruction pipes. The clock throttling mechanism of the present invention permits the instruction pipe to delay operation of its own elements to synchronize them with the outside element. Delay may be introduced for several reasons: to ensure that the processing of the instruction pipe does not exceed the pipe's access to the shared element, and to ensure that the instruction pipe always acts upon valid data from the shared element, even in the presence of significant round-trip communication latency between the instruction pipe and the shared element.

FIG. 1 illustrates an instruction pipe 100 according to an embodiment of the present invention. Instruction pipes 100 may include a plurality of instruction pipestages 110, 120. As is

known, pipestages 110, 120 store data associated with instructions being processed. As instruction data propagates through different pipestages of the instruction pipe 100, the data associated with the instruction may change. This manipulation of instruction data is part of the process of instruction execution.

Although FIG. 1 illustrates only a pair of instruction pipestages 110, 120, an instruction pipe 100 typically includes a cascaded chain of pipestages of a number that may be determined according to conventional design principles. The instruction pipe 100 may include interstitial logic that operates upon instruction data as it propagates through the multiple pipestages 110, 120. For the purposes of this discussion, it is sufficient to illustrate a pair of the pipestages 110, 120 and to explain elements that provide the clock throttling functionality of the embodiment. A pipestage may include additional logic in addition to that shown in FIG. 1 that provides other functionalities. Such additional logic is omitted from this discussion so as not to obscure operation of the clock throttling logic.

The logic shown in FIG. 1 may provide an interface between the pipestages 110, 120 of a first instruction pipe 100 and an RSB (not shown) provided disclosure within a processor. As is known, an RSB may store information relating to two specific types of program instruction: call instructions and return instructions. A call instruction may cause a processor to execute an identified segment of program code. Upon conclusion of the segment, the segment should terminate with a return instruction. The return instruction may cause the processor to return to the call instruction and resume execution with the instruction immediately following the call instruction.

Calls and returns may be nested within other calls and returns. That is, a processor may encounter a first call instruction that causes the processor to execute a first segment of code. The processor further may encounter a second call instruction that causes the processor to execute a second segment of code prior to conclusion of the first segment. The layered relationship of the calls and returns is highly efficient for program designers. In this regard, the characteristics of call and return instructions are well known.

According to this embodiment, when the first instruction pipestage 110 receives instruction information relating to a call instruction, it may store the following information relating to the instruction:

instruction type -- data identifying the instruction as a call instruction,

instruction address -- data identifying an address in an instruction cache from which the call instruction was read,

target address -- data identifying the instruction segment being called, an address in the instruction cache where the program should begin execution.

When the first instruction pipestage 110 receives instruction information relating to a return instruction, it may store data of an instruction type identifying the instruction as a return instruction. The data of a return instruction may change, however, as it propagates to the second pipestage 120. For example, a return address from the RSB or other source may be stored in the second pipestage. In the second pipestage, the data may include not only the instruction type, but also the address identifying the address in the instruction progression to which the processor should return. The second pipestage 120 may receive this address from the RSB.

The instruction pipe 100 may be provided in communication with an RSB (not shown) over communication lines 130, 140. A first set of communication lines 130 may provide an outbound communication link from the first pipestage 110 to the RSB. The first set of communication lines 130 also may be input to a first register 150.

A second set of communication lines 140 may provide an inbound communication link from the RSB to the second instruction pipestage 120 via a second register 160. Outputs of the first and second register 150, 160 each may be input to a selection multiplexer 170. An output of the selection multiplexer 170 may be input to the second pipestage 120.

In an embodiment, the instruction pipe 100 may include a state machine 180 that controls the clock throttling of the pipestage. The state machine 180 may control a read/write controller 190 that interfaces the instruction pipe 100 to the RSB (not shown) and also controls reading and writing of data to the second pipestage 120. The state machine 180 also may control a clock controller 200. The clock controller 200 may cause an input clock signal CLK to be throttled by disabling propagation of the clock signal under control of the state machine 180. In an embodiment, the clock controller 200 simply may be an AND gate.

Consider operation of the instruction pipe 100 in response to a call instruction. When a call instruction is read from the first instruction pipestage 110, the state machine 180 may decode the instruction type data from the first pipestage 110. Based upon the instruction type, the state machine 180 may cause the read/write controller 190 to issue a write command to the

RSB. Address information from the first instruction pipestage 110 may be produced to the RSB over communication lines 130. The address information also may be written to the first register 150.

When a return instruction is read from the first instruction pipestage, again the state machine may decode the instruction based on its instruction type. Again the state machine 180 may control the read/write controller 190. For a return instruction, the read/write controller 190 requests an address to be read from the RSB. The address may be read into the second register 160 over the second communication lines 140.

According to an embodiment, an RSB may operate according to a "read-ahead" policy. The RSB may provide address data from the top of its stack in advance of being requested for the data. Accordingly, after the round-trip communication latency period passes from a previous call or return, the second register 160 should store address data associated with a return instruction at the top of the RSB stack. When the state machine 180 decodes a return instruction, it may cause data from the second register 160 to be read directly into the second instruction pipestage 120 without waiting for a response from the RSB to the read command issued by the read/write controller 190. When the RSB acts upon the read command from the read/write controller 190, the RSB may pop an address from the top of its stack, advances a new address to the top of the stack and pushes it to the second register 160 automatically. This configuration helps to maximize throughput of the instruction pipe.

In an embodiment, the first register 150 stores data associated with a most recent call instruction. According to this embodiment, even if program flow caused a call and a return to occur at a rate that overwhelms the RSB round-trip latency (within less than 5 clock cycles of each other), the return could proceed. The register 150 would store the address of the most recent call instruction. Thus, address information for the return instruction could be stored in the second pipestage. The addition of the first register 150 provides advantages in that it improves throughput of the instruction pipe and leads to a simpler state mechanism 180.

The first instruction pipestage 110 may receive data relating to an instruction to be processed. Instruction data may identify an instruction type. For different instruction types, the instruction data may differ. For a call instruction, in addition to an instruction type identifying the instruction as a "call," the instruction may include an address of an instruction that represents a return point from the call.

In an embodiment, the state machine 180 controls the multiplexer 170 to cause return data from one of the first or second registers 150, 160 to be stored in the second instruction pipestage 120.

Typically, within an instruction pipe 100, data of a single instruction propagates through one pipestage for each clock cycle. Thus, during operation, data may be read out of the first pipestage 110 at a first clock cycle. According to an embodiment, type information from the instruction may be input to the state machine 180. As described herein, the state machine may distinguish between call instructions, return instructions and all other instructions.

According to an embodiment, the state machine 180 may be adapted to conform the operation of the instruction pipe 100 to timing limitations of the RSB. Typical timing limitations include:

- **RSB availability.** As noted, the RSB may be shared by several elements within a processor. Thus, there may be a predetermined limit to the frequency with which an instruction pipe may issue requests to the RSB. Requests issued in excess of this limit may be dropped by the RSB.
- **Communication latency from the instruction pipe to the RSB.** As noted, the RSB may be provided in one instruction pipe and field requests from another instruction pipe. A physical separation between the RSB and the other instruction pipe may impose a predetermined round-trip communication latency to communications between them.

Different embodiments may generate different timing limitations. According to an embodiment, the different timing limitations may be predetermined and programmed into the state machine 180. Thus, the state machine may monitor instructions as they propagate from first pipestage 110 to the second pipestage 120. The state machine 180 may determine if a sequence of program instructions may occur that make it possible to create invalid data in the second pipestage 120. If so, the state machine 180 may throttle the clock in the instruction pipe 100 to suspend its operation until valid data is available.

According to an embodiment, the clock controller 200 may generate a local clock signal LCLK. The local clock LCLK may be input to the two pipestages 110, 120 and to every pipestage preceding the first pipestage 110 in the instruction pipe 100. There is no requirement that the local clock LCLK be input to any pipestages following the second pipestage 120. Thus, when the state machine 180 stalls the instruction pipe 100, it need only stall a portion of the instruction pipe 100 that precedes the stalled call or return instruction.

FIG. 2 illustrates operation of the state machine according to an embodiment of the present invention. To better illustrate the operation of the embodiment, FIG. 2 illustrates operation of a system using specific examples for the two timing limitations discussed above:

- The instruction pipe may issue only one request to the RSB every two clock cycles.
- Round-trip communication with the RSB takes five cycles.

Of course, the principles of the present invention find application with systems having other timing limitations.

The state diagram 300 of FIG. 2 illustrates several states 310-440 of operation for the state machine 180. The state machine 180 may transition among each of the various states once per cycle of the master clock signal CLK (FIG. 1). In FIG. 2, text provided adjacent to each of the arrows represents a condition that causes the state machine to advance from a first state to a second state.

At initialization, the state machine 180 may start in an idle state 310. As instructions propagate through the instruction pipe, the state machine 180 may classify instructions into three types for the purposes of clock throttling: a call instruction, a return instruction or an "other" instruction (not a call, not a return). The return instruction may cause the state machine to advance to a state 320 and issue the return. The call instruction may cause the state machine to advance to another state 330 and issue the call. Any other instruction may be processed according to normal procedures; the state machine 180 may remain at the idle state 310 for the purposes of clock throttling.

From state 320, the state machine 180 may classify an instruction as a return, a call or other. If a second return instruction occurs (back-to-back returns), the state machine 180 may begin a multi-clock stall, represented by states 340-370. Using the example of a 5 cycle round-trip communication delay, the clock stall would occur for 5 cycles. Again, other embodiments may be appropriate for different round-trip communication latencies. From state 320, the state machine 180 may progress in sequence through each of the stall states 340-370 regardless of the instructions that follow the second return. At the conclusion of the stall at state 370, the state machine returns to state 320 and issues the second of the back-to-back returns. Thereafter, the state machine determines the type of the instruction following the second return and processes this new instruction based on its instruction type.

From state 320, a next instruction may be a call instruction (a return, followed immediately by a call). In this case, the state machine 180 may advance to state 380 and stall the clock. Thereafter, the state machine may advance to state 390 and issue the call instruction. This single-cycle stall satisfies the timing limitation that requires sequential communications with the RSB to be separated by at least two cycles. In this case, although the call instruction arrives at the state machine 180 immediately after the return that preceded it, the call instruction will be delayed by a cycle to ensure that the communications with the RSB satisfy the 2 cycle timing limitation.

From state 390, the state machine 180 examines a next instruction from the first pipestage 110. Again, the instruction may be classified as a call, return or other. If the state machine 180 determines that the new instruction is a return, the state machine 180 may advance to state 360 and stall the instruction pipe. This response is appropriate because the instruction sequence (return-call-return) implicates both timing limitations. Without clock throttling, the instruction sequence would violate not only the 2 cycle limitation governing sequential communication with the RSB but also the 5 cycle limitation governing round-trip communications with the RSB. By advancing to state 360, the state machine will satisfy both timing limitations. It will wait another three cycles (states 360, 370, 320) before issuing the return instruction.

From state 390 the state machine may determine that the next instruction is a call instruction. In this case, the state machine may advance to state 400 and stall the clock for a single cycle. Thereafter, the state machine may advance to state 330 and issue the call instruction.

From state 390, the state machine may determine that the next instruction is neither a call nor a return. In this case, the state machine 180 may advance to state 410. The state machine 180 may permit the instruction to proceed. At state 410, the state machine processes a next instruction. For a call instruction, no timing requirements prevent the instruction from being issued; the state machine 180 may advance to state 330 and issue the call instruction. For an "other" instruction, the state machine may advance to the idle state 310. Execution of the "other" instruction will cause the state machine to be removed from the return issued at state 320 by the minimum five cycles; therefore, the state machine may return to the idle state.

If the state machine detects a return at state 410, the state machine may advance to state 370. The new instruction is part of a sequence, return-stall-call-other-return. Advancing from state 410 to state 370 is appropriate to ensure that the 5 cycle round-trip latency is maintained. Thus, the second return will be stalled for a clock cycle. Thereafter, the state machine may advance to state 320 and issue the return.

As noted, when the state machine is at the idle state 310 and detects a call instruction, it may advance to state 330 and issue the call. Thereafter, the state machine may classify a new instruction following the call. If the new instruction is itself a call instruction (back-to-back calls), the state machine may advance to state 400 and stall the new instruction. After the single-cycle stall, the state machine may return to state 330 and issue the second call.

If, at state 330, a next instruction is a return, the state machine may advance to state 370 and stall the instruction. Following state 370, the state machine may advance to state 320 and issue the return instruction.

From state 330, any instruction classified at "other" may cause the state machine to return to the idle state. The instruction may proceed.

From state 320, if the state machine classifies a new instruction as other, it may permit the new instruction to proceed (state 420). From state 420, a return instruction may cause the state machine to advance to state 350, an appropriate state in the multi-cycle stall that must be maintained for successive return instructions. From state 420, a call instruction may cause the state machine to advance to state 390. The call may be issued. From state 420, any other instruction may cause the state machine to advance to a state 430; the other instruction is permitted to proceed.

From state 430, a call instruction may cause the state machine to advance to state 330 and any other instruction may cause the state machine to advance to a state 440. No timing limitation prevents either the call instruction or other instruction from being processed. A return instruction, however, may cause the state machine to advance from state 430 to state 360. A remainder of the multi-cycle delay for successive returns must be completed prior to issuing the second return at state 320.

From state 440, a return instruction may cause the state machine to advance to state 370 for a single-cycle stall. A call instruction, however, may be issued; the state machine,

therefore, may advance to state 330. Any other instruction also may be issued; the state machine 180 thereafter returns to the idle state 310.

The principles of the present invention find application with other embodiments of instruction pipes such as those that independently process multiple code streams. In common operating systems, these multiple streams are called “threads”. FIG. 1 illustrates in phantom a second pair of registers 210, 220 that may be used in a multi-stream instruction pipe. Register 210 may be provided in communication with the first communication path 130 and register 220 may be provided in communication with the second communication path 140.

In this embodiment, as instructions propagate through the instruction pipe 100, the instructions carry data identifying the stream to which the instruction belongs. Of the pair of registers 150, 210 connected to the first communication path 130, a first register 150 may store call addresses associated with one stream and a second register 210 may store similar addresses associated with another stream. So, too, with the registers 140, 220 connected to the second communication path 140. A first register 140 may store call addresses associated with the first stream and a second register 220 may store similar addresses associated with the second stream.

A multi-stream instruction pipe may operate under the same timing restraints as a “uni-stream” instruction pipe. Accordingly, operation of the state machine 180 need not change from the embodiments described above. However, as the state machine 180 controls the multiplexer 170 to read data from one of the registers 150-160, 210-220 to the second instruction pipestage 120, it performs its selection based not only on the timing of the instructions but also the stream from which the instruction originated.

FIG. 3 is a block diagram illustrating an execution unit of a processor having two instruction pipes 510, 520. The first instruction pipe 510 may include a first plurality of instruction pipestages 530 and the second instruction pipe 520 may include a second plurality of pipestages 540. Thus, the two instruction pipes 510, 520 provide for parallel execution of program instructions.

In the embodiment shown in FIG. 1, a first instruction pipe 510 includes an RSB 530. The second instruction pipe 520 is provided in communication with the RSB 530 via an interconnect 540. Thus, communication between the second instruction pipe 520 and the RSB 530 may be affected by any latency imposed by the interconnect 540.

According to an embodiment, an RSB 530 may communicate with multiple stages of a single instruction pipe. For example, the RSB 530 accepts two communication links 550, 560 from the second instruction pipe 520. In this embodiment, each communication link 550, 560 may be treated by the RSB 530 as a different entity. Thus, each portion of an instruction pipe
5 may have access to the RSB 530 on a pro rata basis based on the number of communication links into the RSB 530 rather than just an absolute number of instruction pipes in the processor.

Although FIG. 3 illustrates the RSB provided within an instruction pipe 510, the principles of the present invention are not so limited. Thus, the present invention accommodates alternate embodiments such as those where the RSB 530 would be provided as a separate circuit
10 independent from the first instruction pipe 510. In the alternate embodiment, communication links 570, 580 from the first instruction pipe 510 to the RSB 530 would pass through a second interconnect (not shown).

Several embodiments of the present invention are specifically illustrated and described herein. However, it will be appreciated that modifications and variations of the present invention
15 are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

WE CLAIM:

1. An instruction pipe control method comprising:

reading a new instruction from an instruction pipestage,

determining, with reference to other instructions read previously from the instruction

pipestage, whether valid data associated with the new instruction can be written to a next
instruction pipestage and,

stalling processing of the new instruction until valid data associated with the new
instruction can be written to the next instruction pipestage.

2. The method of claim 1, wherein, if the new instruction is a return instruction, the
determining includes determining whether a return address is available within the instruction
pipe.

3. The method of claim 1, wherein, if the new instruction is a return instruction, the
determining includes determining whether sufficient time has expired from an earlier return
instruction for a return address to be received from an external resource.

4. The method of claim 1, wherein, if the new instruction is a call instruction, the
determining includes determining whether immediate processing of the call instruction would
exceed a predetermined access rate associated with a shared resource.

5. The method of claim 4, further comprising, after the stalling terminates, storing a return
address associated with the call instruction both locally and in a shared resource.

6. The method of claim 1, wherein the stalling stalls the instruction pipestage and all other
instruction pipestages before it in the instruction pipe.

7. An interface method for an instruction pipe that shares access to an external resource,
comprising:

reading a new instruction from an instruction pipestage,

if the new instruction requires access to the external resource, determining with
reference to other instructions read previously from the instruction pipestage, whether
immediate processing of the new instruction would cause the instruction pipe to exceed an
access allocation for the instruction pipe, and

if so, stalling the new instruction.

8. The method of claim 7, further comprising, if the new instruction is a call instruction, after the stalling terminates, storing a return address associated with the call instruction both locally and in a shared resource.

9. The method of claim 7, wherein the stalling stalls the instruction pipestage and all other instruction pipestages before it in the instruction pipe.

10. A method for interfacing an instruction pipe with an external resource characterized by a predetermined round-trip communication latency period, the method comprising:

reading a new instruction from an instruction pipe stage,

determining, with reference to other instructions read previously from the instruction pipestage, whether valid data associated with the new instruction is available to the instruction pipe prior to expiration of the round-trip communication latency period,

if not, stalling processing of the new instruction until the round-trip communication latency period expires.

11. The method of claim 10, further comprising

determining whether the new instruction requires access to the external resource in excess of an access allocation for the instruction pipe, and

if so, stalling the new instruction.

12. The method of claim 10, wherein the stalling stalls the instruction pipestage and all other instruction pipestages before it in the instruction pipe.

13. In an instruction pipe, a clock throttling mechanism provided between a pair of instruction pipestages, comprising:

a state machine coupled to an output of a first instruction pipestage,

a clock control circuit having an input for a system clock signal and having an output for a modified clock signal, the output coupled to the pair of instruction pipestages, the clock control circuit controlled by the state machine.

14. The clock throttling mechanism of claim 13, further comprising a read/write controller under control of the state machine and having an output for controlling writes to the second instruction pipestage.

15. The clock throttling mechanism of claim 13, further comprising:

a first register coupled to the first instruction pipestage,

a second register,
a selector coupled to the first and second registers and having an output coupled to the second instruction pipestage.

16. The clock throttling mechanism of claim 15, wherein the selector is controlled by the state machine.

17. Execution logic for a processor, comprising:
a first instruction pipe, comprising a first plurality of cascaded pipestages, and
a return stack buffer provided in communication with at least one of the first pipestages;
and

a second instruction pipe, comprising:
a second plurality of cascaded pipestages, at least one of the second pipestages
provided in communication with the return stack buffer, and
clock throttling logic coupled to the at least one second pipestage.

18. The execution logic of claim 17, wherein the clock throttling logic comprises:
a state machine coupled to an output of the one pipestage from the second plurality,
a clock control circuit having an input for a system clock signal and having an output for
a modified clock signal, the output coupled to the one pipestage, the clock control circuit
controlled by the state machine.

19. The execution logic of claim 17, further comprising, in the first instruction pipe, second
clock throttling logic that comprises:
a state machine coupled to an output of the one pipestage from the first plurality,
a clock control circuit having an input for a system clock signal and having an output for
a modified clock signal, the output coupled to the one pipestage, the clock control circuit
controlled by the state machine.

20. The execution logic of claim 17, wherein additional instruction pipestages from either the
first or the second instruction pipe are provided in communication with the return stack buffer,
the additional instruction pipestages also provided with additional clock throttling logic.

ABSTRACT

Multiple instruction pipes may share access to a common resource. The instruction pipes may be able to process data from the resource at a faster rate than the rate at which the resource may furnish data to the instruction pipe. An instruction pipe therefore, when it reads a new instruction from an instruction pipestage, determines whether valid data associated with the new instruction can be written to a next instruction pipestage. The instruction pipe stalls processing of the new instruction until valid data associated with the new instruction can be written to the next instruction pipestage.

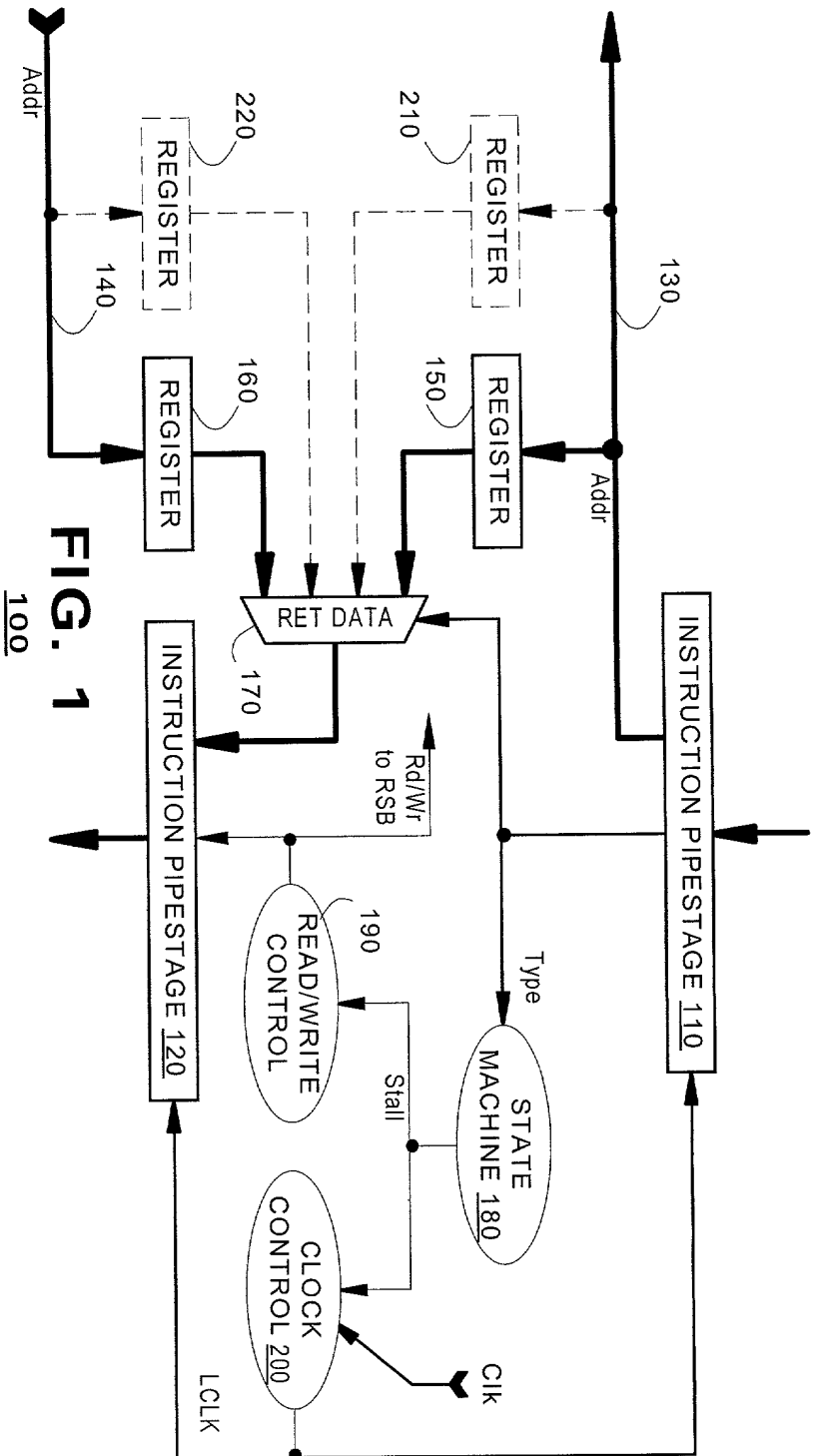
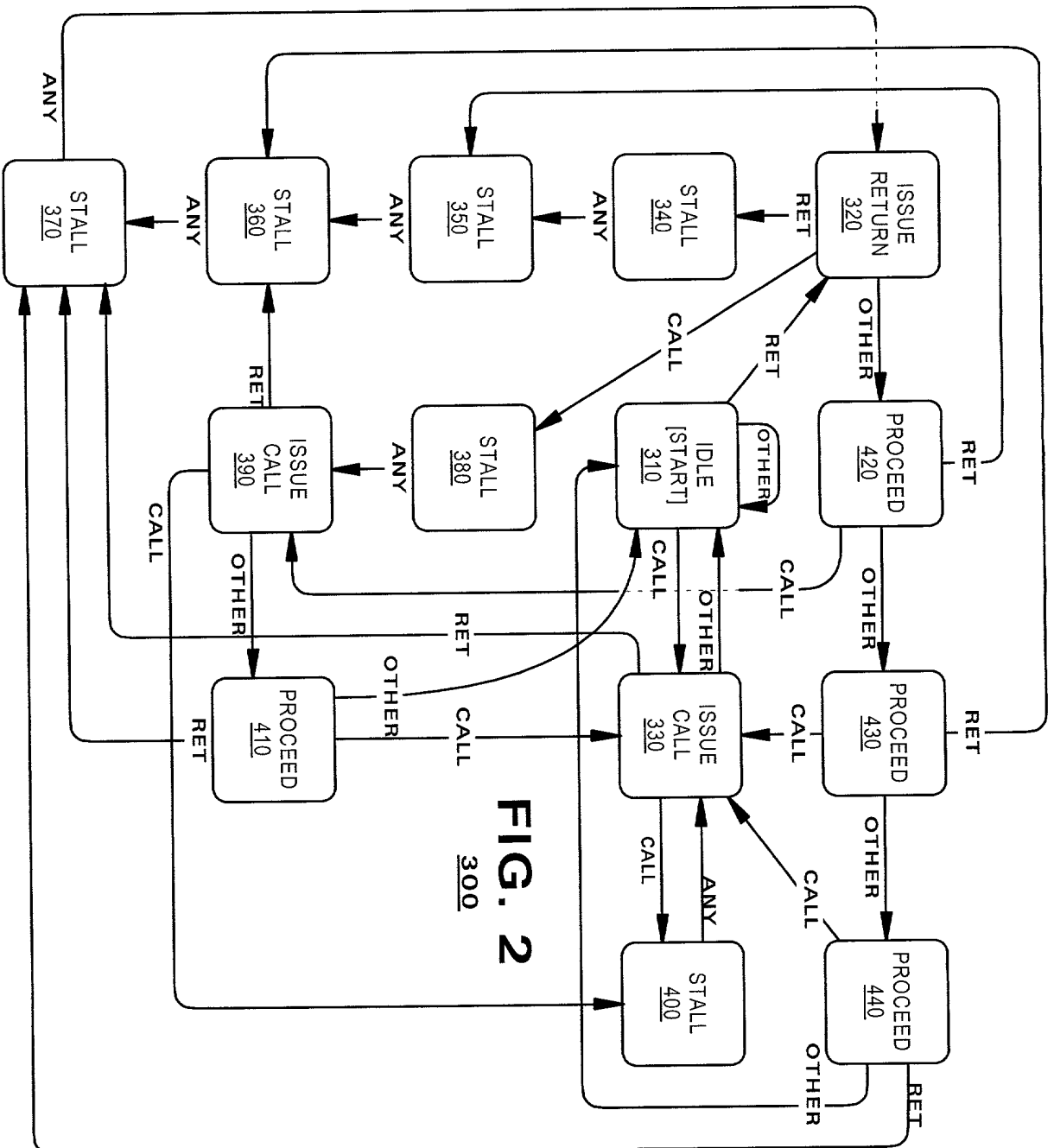


FIG. 1

100



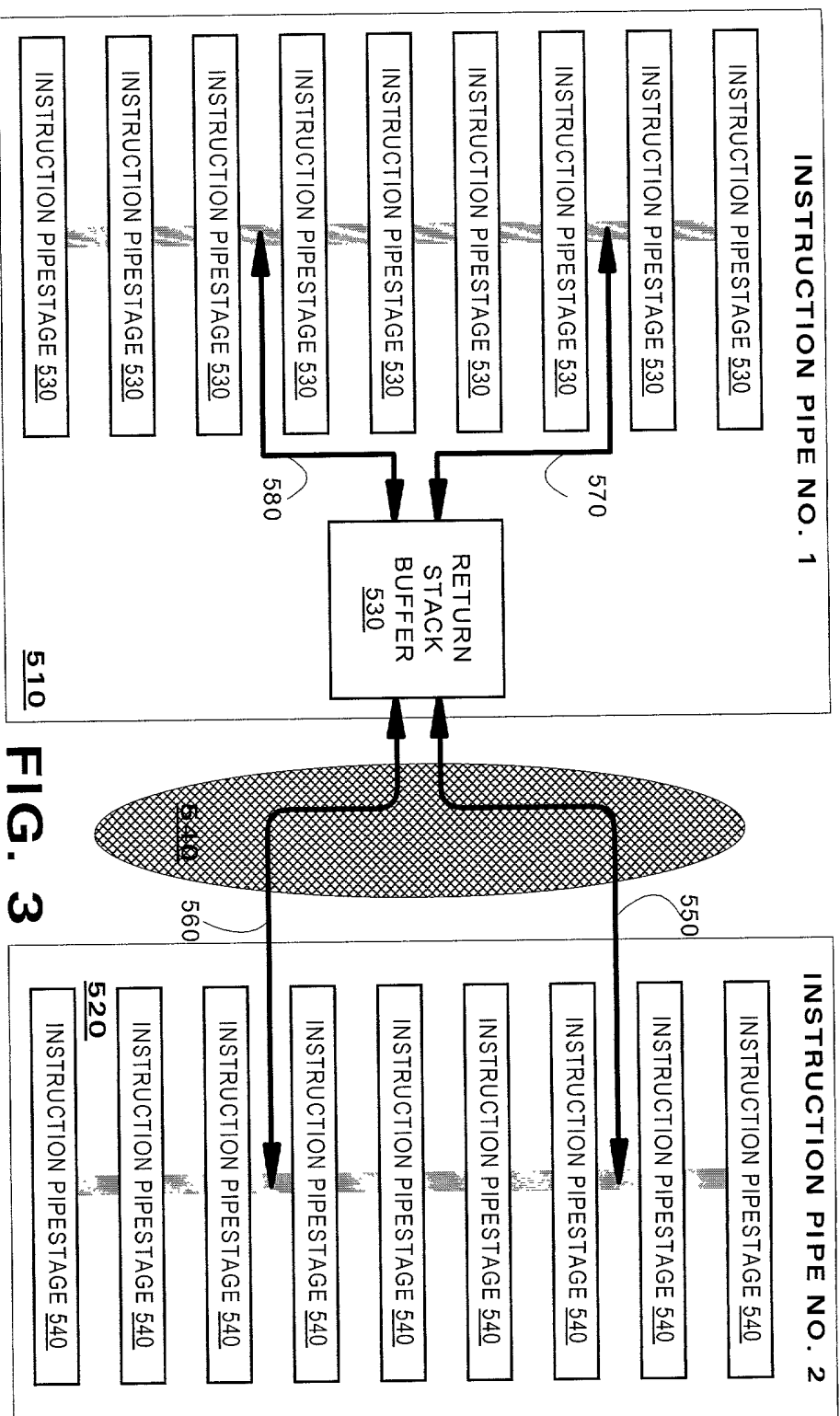


FIG. 3

PATENT**Docket No. 2207/8613****DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

CLOCK THROTTLE FOR INSTRUCTION PIPE

the specification of which is attached hereto unless the following is entered:

was filed on	as United States Application Number or PCT International Application Number	and was amended on (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR §1.56.

PRIOR FOREIGN APPLICATION(S)

I hereby claim foreign priority benefits under 35 USC §119(a-d) or §365(b) of any foreign application(s) for patent or inventor's certificate, or §365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below any foreign application(s) for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed:

Application Number	Country	Filing Date (day/month/year)	Priority Not Claimed

PROVISIONAL APPLICATION(S)

I hereby claim the benefit under 35 USC §119(e) of any United States provisional application(s) listed below:

Application Number	Filing Date

PRIOR UNITED STATES APPLICATION(S)

I hereby claim the benefit under 35 USC §120 of any United States application(s), or §365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 USC §112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR §1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

Application Number	Filing Date	Status (patented, pending, abandoned)

POWER OF ATTORNEY

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Paul H. Heller (Reg. No. 21,074); John C. Altmiller (Reg. No. 25,951); Shawn W. O'Dowd (Reg. No. 34,687); Robert L. Halls, Jr. (Reg. No. 39,702) of KENYON & KENYON with offices located at 1500 K Street NW, Suite 700, Washington, DC, 20005-1257, telephone (202) 220-4200, and at 333 W. San Carlos Street, Suite 600, San Jose, CA, 95110-2711, telephone (408) 975-7500;

and Alan K. Aldous (#31,905); R. Edward Brake (#37,784); Ben Burge (#42,372); Jeffrey S. Draeger (#41,000); Cynthia Thomas Faatz (#39,973); John N. Greaves (#40,362); Seth Z. Kalson (#40,670); David J. Kaplan (#41,105); Peter Lam (#44,855); Charles A. Mirho (#41,199); Leo V. Novakoski (#37,198); Thomas C. Reynolds (#32,488); Kenneth M. Seddon (#43,105); Mark Seeley (#32,299); Steven P. Skabrat (#36,279); Howard A. Skaist (#36,008); Gene I. Su (#45,140); Calvin E. Wells (#43,256); Raymond J. Werner (#34,752); Robert G. Winkle (#37,474); and Charles K. Young (#39,435) of INTEL CORPORATION.

331541

PATENT
Docket No. 2207/8613

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION (Cont.)

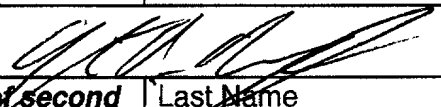
Direct telephone calls to:

John Altmiller
(202) 220-4200

Send correspondence to:

KENYON & KENYON
1500 K Street, NW, Suite 700
Washington, DC 20005-1257

I hereby declare that all statements made herein of my own knowledge are true and all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under §1001 of Title 18 of the United States Code and that such willful statements may jeopardize the validity of the application or any patent issuing thereon.

Full name of first or sole inventor	Last Name DOUGLAS	First Name Jonathan	Middle Name P.
Residence	City Portland	State or Country Oregon	Country of Citizenship U.S.A.
Post Office Address	Street 10500 NW Cornell Road	City Portland	State or Country & Zip Code Oregon 97229
Signature 		Date 6/19/00	
Full name of second inventor	Last Name	First Name	Middle Name
Residence	City	State or Country	Country of Citizenship
Post Office Address	Street	City	State or Country & Zip Code
Signature		Date	
Full name of third inventor	Last Name	First Name	Middle Name
Residence	City	State or Country	Country of Citizenship
Post Office Address	Street	City	State or Country & Zip Code
Signature		Date	